User Manual

TE1014

PXI Fourteen Slot Test Platform

CE

Team Solutions, Inc.
27071 Cabot Road, Suite 124
Laguna Hills, CA 92653

Tel (949)348-7766 Fax (949)348-7776
Email : sales@team-solutions.com
Web : http://www.team-solutions.com

TE1014 User Manual

Description: *TE1014 PXI 14 Slot Chassis User Manual*

Revision History				
Rev	Date	Description	Written by:	Checked by:
1.0	27 Feb 2002	Initial Release	CC	CC
1.1	16 May 2002	New Company Info	CC	CC

Copyright © 2002 Team Solutions, Inc. All rights reserved. Products and company names used in this document are trademarks of their respective companies.
The PXI™ logo and Product name are trademarks of the PXI Systems Alliance.

Team Solutions, Inc. reserves the right to change specifications and contents of this document without prior notice.

Warranty.

The TE1014 is warranted against hardware defects in materials and workmanship for a period of one year from the date of delivery as noted on the delivery and/or shipping documentation. Team Solutions, Inc. is not responsible for any damage caused to its product, or any other equipment caused by misuse of this product or damage caused to this product by the failure to follow the instructions of Team Solutions, Inc., or problems caused by other third party equipment, supplies, services, modification to supplied equipment, etc. Team Solutions, Inc. will repair/replace at their option any equipment that proves to be defective during the warranty period. This warranty applies to parts and labor. Team Solutions, Inc. cannot be held responsible for any consequential loss due to defects in supplied product.

Year 2000

Each of the products supplied by Team Solutions, Inc. are designed using Y2K Compliant tools.

Team Solutions, Inc. is not responsible for any damage or loss caused to its product, or any other equipment caused by the use of Team Solutions product with any other product or equipment that is not certified as Year 2000 compliant.

Warning

Products designed by Team Solutions, Inc. are not tested for use in medical or clinical applications.

The TE1014 series of PXI chassis is designed for use; only under the power supply and operating environment conditions as defined in section 1.2 of this manual.

Table of Contents

1. INTRODUCTION 7

1.1 COMPANY DESCRIPTION. 7
1.2 PRODUCT DESCRIPTION. 7
1.3 MODELS AND OPTIONS AVAILABLE. 9
1.4 REGULATORY NOTICES 9
1.4.1 FCC Notice (U.S. Only) 9
1.4.2 IC Notice (Canada Only) 9
1.4.3 VCCI Notice (Japan Only) 10
1.5 SAFETY NOTICES 10
1.6 POWER CORD 11

2 2. RESOURCES REQUIRED 11

2.1 HARDWARE. 11
2.2 SOFTWARE. 11

3 HARDWARE SET-UP 12

3.1 CHECKLIST 12
3.2 POWER SWITCH AND INDICATIONS. 12
3.3 MOUNTING 13
3.4 REPLACING A FUSE 13
3.5 CLEANING FILTERS 14
3.6 AIR FLOW 14

4 SOFTWARE INSTALLATION 16

4.1 BRIDGE DRIVER INSTALLATION. 16
4.2 OPTION DRIVER INSTALLATION. 17
4.3 INSTALLING OTHER PXI/COMPACTPCI MODULES 17
4.4 VERIFYING INSTALLATIONS 17
4.4.1 Bridge Card 17
4.4.2 Other PXI Modules 18

5 OPERATION AND CONNECTIVITY 20

5.1 TE1014 20
5.1.1 Chassis topology 20
5.1.2 Peripheral slot identification 22
5.1.3 Star trigger 24
5.1.4 Local Bus 25
5.1.5 PXI Trigger Bus 25
5.1.6 10MHz system clock 26
5.1.7 Chassis Monitor Panel 27

6 APPENDICES 31

6.1 APPENDIX A – TEAMBRIDGE.DLL CALLS 31
6.2 APPENDIX B - PXI TRIGGER BUFFER 50
6.3 APPENDIX C - PROBLEM SOLVING 52
6.4 APPENDIX D - EXAMPLE PROGRAM 55
6.5 APPENDIX E – SLOT PINOUT TABLE 56

7 NOTES 59

1. Introduction

1.1 Company description.

Established in 1999 and located in Southern California, TEAM's mission is to provide its customers with state-of-the-art, PC-Based, Test & Measurement instrumentation and data acquisition (DAQ) devices. TEAM's products are offered in ISA, PCI, PCMCIA, External (Parallel/USB driven), CompactPCI and PXI configurations.

1.2 Product description.

Team Solutions, Inc.'s TE1014 is a 14 slot, 19 inch rack-mountable chassis, which fully complies with the PXI and Compact PCI standards. All slots have full PXI functionality. Separate forced-air cooling is provided for the power supply and PXI slots. Fan noise is minimized by varying fan speed according to system temperature. The TE1014 model includes features such as temperature and PSU monitoring.

The expanded slot capability of the unit provides system integrators with a single-box solution, thus avoiding synchronization, integration and cost issues associated with multi-platform solutions. With its compact case size of 4U / 19 inch, the TE1014 provides the small footprint required for today's ATE and Data Acquisition applications.

- Accommodates PXI controller, Star Trigger, and up to 13 PXI Cards.
- Integral PCI/PXI Bridge Cross (X) Point Bridge.
- Full PXI functionality on all slots.
- 19 Inch Rack mount case (4U).
- Separate forced air-cooling for PSU and PXI slots.
- Variable Speed Fans (PXI slots) minimize noise.
- Temperature / PSU Monitoring.

Specifications

Backplane Configuration	1 Embedded Controller (16HP) or MXI-3 Slot. 1 PXI / CompactPCI or Star Trigger Controller slot. 12 PXI / CompactPCI peripheral slots.	
Bridge	X-Point Palatte Bridge. (Bridges primary and secondary Trigger Bus signals via a matrix. Capable of routing signals back onto different triggers on the source side of the backplane). Trigger bus matrix propagation delay: Approx 25nS	
System Power	Type:	350 W auto-sensing supply

	Input:	120-240 V~, 50-440 Hz
	Outputs:	5V DC @ 35A, 3.3V DC @ 35A +12V DC @ 10A, - 12V DC @ 4A
	Fuse:	5 Amp, Bussmann Anti-Surge S505 (T), 5*20 mm. T5AH/250~
Available Slot Current (As per PXI Spec 2.0)	Controller Slot:-	5V @ 6A 3.3V @ 6A +12V @ 0.5A -12V @ 0.1A
	Each Peripheral Slot:-	5V @ 2A 3.3V @ 0A +12V @ 0.5A -12V @ 0.1A

Cooling PXI Slots 2 thermistor controlled variable speed fans @ 100 CFM (max) each
PSU Bay : Single 25 CFM in-line fan

Voltage Monitor 3.3V DC, 5V DC, 12V DC, -12V DC.

Temperature Monitor Over PXI slot 5, PXI slot 10, Controller Slot and in PSU bay.

Dimensions H x W x D : 4U (7") **x** 19" **x** 16.54"
177.8 mm **x** 482.6 mm **x** 420 mm

Standards IEC 61326, IEC 1010, IEC1010-1: 1990 and amendments 1&2, EN50082-1, EN55011, EN61010 PXI Spec Rev 2.0, UL3111-1: 1994, CAN/CSA22.2No. 1010-1

1.3 Models and Options Available

TE1014 – PXI 14 Slot PXI Chassis.

Accommodates PXI controller, Star Trigger, and up to 13 PXI Cards.
Integral PCI/PXI Bridge, Full PXI functionality on all slots
19" Rack mount case, Separate forced air-cooling for PSU and PXI slots
Temperature Controlled Variable Speed Fans (PXI slots) minimize noise
System Temperature and Power Supply Monitoring.
Trigger Bus crosssegment X-Point connecitivity

Software provided includes application software drivers - VISA®, LabVIEW® and LabWindows®

Additional support is provided for Visual Basic, C and C++ through DLL's and example programs and Borland C++

1.4 Regulatory Notices

1.4.1 FCC Notice (U.S. Only)

All models of the Team Solutions, Inc. TE1014 series PXI Chassis are classified by the FCC as Class A digital devices and as such the following FCC notices apply:

WARNING: Changes or modifications to this unit not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the manufacturer's instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at personal expense.

Shielded cables must be used with this unit to ensure compliance with the Class A FCC limits.

1.4.2 IC Notice (Canada Only)

All models of the Team Solutions, Inc. TE1014 series PXI Chassis are classified by Industry Canada (IC) ICES-003, Issue 2 as Class A digital devices. Devices that are tested and comply with Class A FCC Specifications also comply with the Canadian Specification following the harmonisation of Canadian Regulation with existing FCC regulations effective January 31, 1989. As such the following notices apply:

This digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le present appareil numerique n'emet pas de bruits radioelectriques depassant les limites applicables aux appareils numeriques de la class A prescrites dans le Reglement sur le brouillage radioelectrique edicte par le ministere Des Communications du Canada.
CE Notice

Marking by the **CE** symbol indicates compliance of all models of the Team Solutions, Inc. TE1014 series PXI Chassis to the EMC directive of the European Community. Such marking means that these TE1014 chassis meet technical standard EN55022 - Limits and Methods of Measurements of Radio Interference Characteristics of Information Technology Equipment (ITE) as a Class A device. As such, the following notices apply:

WARNING: This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

1.4.3 VCCI Notice (Japan Only)

All models the Team Solutions, Inc. TE1014 series PXI Chassis are classified by the Voluntary Control Council for Interference (VCCI) as Class 1 information technology equipment (ITE) and as such the following VCCI notices apply:

Class 1 Notice

This equipment complies with the limits for a Class 1 digital device (devices used in commercial and/or industrial environments) and conforms to the standards for information technology equipment (ITE) that are set by the Voluntary Control Council for Interference for preventing radio frequency interference in commercial and/or industrial areas.

Consequently, when used in a residential area or in an area adjacent to a residential area, this equipment may cause radio interference with radio and television receivers or other communications equipment.

To ensure that such radio interference does not occur, it is important to install and use this equipment in accordance with the manufacturer's instruction manual.

1.5 Safety Notices

This equipment is designed to be safe at least under the following conditions:
Indoor use, altitude up to 2000m, temperature 5 Degrees Centigrade to 40 Degrees Centigrade, maximum relative humidity 80% for temperatures up to 31 Degrees Centigrade decreasing linearly to 50% relative humidity at 40 Degrees Centigrade, mains supply fluctuations not to exceed +/- 10% of the nominal voltage, transient overvoltages according to Installation Category II and Pollution degree 2 in accordance with IEC 664

1.6 Power Cord

Only use power cords that are approved for use in electrical equipment and are appropriate for the country of use.

2. Resources required

2.1 Hardware.

The TE1014 range of PXI Chassis can be used with any 3U PXI cards and CompactPCI cards which comply with the PXI Revision 1.0 specification (or greater) announced by National Instruments in August 1997 and with the CompactPCI specification of the PCI Industrial Computer Group (PICMG).

It is recommended that Windows 98, Windows NT 4.0, Windows2000 or greater operating systems be used with this chassis. The minimum recommended memory requirement for a Windows 98 operating system is 32 Mbytes of RAM and for a Windows NT/2000 operating system is 64 Mbytes of RAM.

Note: Windows 95 or lower will not support multiple bridging so cannot be used with the TE1014 series chassis.

2.2 Software.

The TE1014 option modules use NI VISA drivers. Before these drivers are installed, a current version of National Instrument VISA (NI-VISA Version 2.01 or greater) must be installed. If this has not already been installed on your system you can install it from the CD supplied with your chassis, or the latest version can be down loaded directly from the National Instruments web site at www.ni.com.

The CD supplied with the TE1014 chassis contains software drivers, sample programs and control panels that allow the user/integrator to avail of the extra functionality supplied with the various different options of the chassis such as System Monitoring and Auxiliary I/O.

3 Hardware Set-up

3.1 Checklist

Before powering up the TE1014 Chassis,

- Check to ensure that there has been no damage caused by the shipping of the unit.
- Check that no screws or panels have come loose.
- Ensure that you connect the unit to a voltage and frequency range within those listed in the system power input specifications in section 1.2 of this manual.
- Ensure that the chassis is correctly grounded at all times when installed or/and set-up for use, even when switched off.

3.2 Power switch and indications.

There are two switches on the TE1014 Chassis. One is on the rear of the chassis and the other is on the front of the chassis.

The switch on the rear of the chassis is a Mains voltage switch and is used to connect mains voltage to the chassis power supply. When switched on, the AC OK LED on the front of the chassis will be on and the power supply fan will be running.

The switch on the front of the chassis is a low voltage DC switch connected to the power supply module control circuit and controls the power supply DC modules. When switched off, there is NO DC voltage applied to the PXI Backplane. When switched on, DC voltages of +5V, +3.3V, +12V and –12V will be applied to the backplane, the rear panel cooling fans will run and the **DC OK** LED will be on.

The appliance coupler or the power cord must be accessible for disconnect in the case of emergency.

3.3 Mounting

Ensure that the unit is disconnected from any mains supply.

As shipped, the unit is equipped with four feet for bench top use. If it is required that the chassis needs to be rack mounted, the feet need to be removed. To remove the feet, unclip the feet covers. Remove the screws one at a time by removing the mounting screw using a **T15** Torx driver. Replace the mounting screw.

WARNING: When mounting in a 19-inch rack, ensure that the chassis rear is supported. The front mounting flanges MUST NOT be used for sole support. Mounting rails suitable for this purpose are supplied by most of the major enclosure manufacturers.

3.4 Replacing A Fuse

To replace the fuse on a TE1014 chassis first isolate the chassis from all mains power sources. With a flat headed screwdriver ease out the fuse holder located below the ON/OFF switch on the rear panel of the chassis.

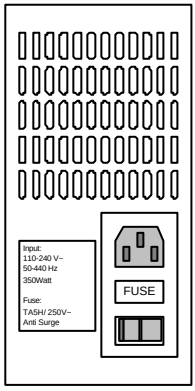


Figure 3.1 - Front Panel

Replace the blown fuse with a 5 Amp, Bussmann Anti-Surge S505 (T) fuse, dimensions 5 mm x 20 mm. Ease the fuse holder back into the switch. Unit can now be reconnected and powered up.

3.5 Cleaning Filters

There are two filters on a TE1014, both of which are located at the

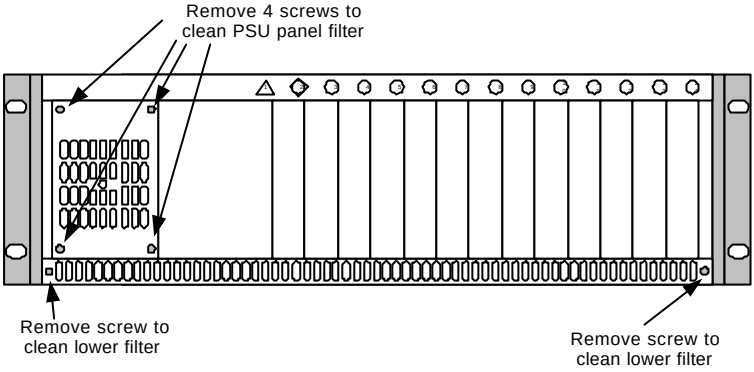


Figure 3.2 – Chassis Front View.

front of the unit to allow for easy access. To access the filter on the Power supply unit, remove the four corner flat head shoulder screws from the front panel. Care should be taken not to damage the LED's or DC switch mounted on the PSU panel. To extract the filter remove the screw in the centre of the plate.

To extract the filter in the bottom panel of the unit, remove the two cross head screws at each end.

Filters may be cleaned using warm soapy water or replaced with a filter replacement kit.

3.6 Air flow

The TE1014 Chassis card frame is cooled with two 50-100 CFM variable speed fans. The air is brought in through the grille in the front of the chassis, sucked up through the card frame and exhausted through the rear.

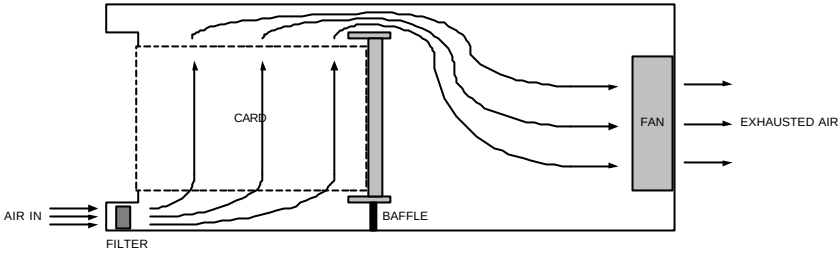


Figure 3.3 - Air Flow Model.

To ensure that maximum airflow is directed through the card frame area, blank panels MUST be used to cover slots where there are no cards installed.

4 Software Installation

4.1 Bridge Driver Installation

Insert the CD supplied with the unit into a CD drive that is accessible by the PXI Controller that is used with the chassis. USB Port or Parallel Port CD drives are suitable for this. If these are not available, it is possible to span the required files across a number of floppy disks using a utility such as Microsoft Backup.

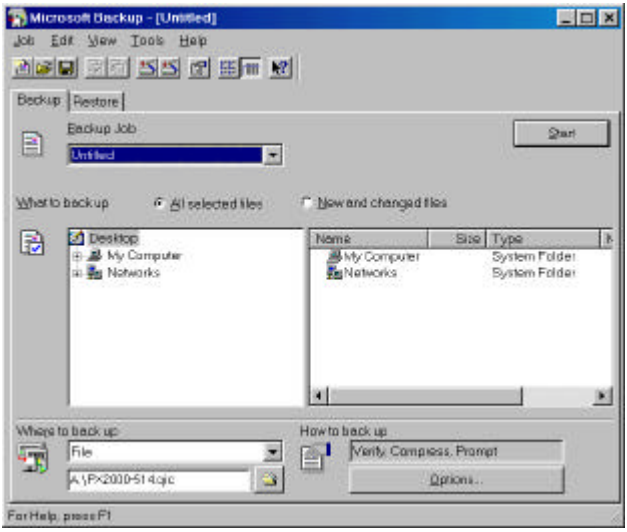


Figure 4.1 - Microsoft Backup

Locate the directory on the CD, which refers to TE1014 / Bridge Driver and run the SETUP.EXE program. This file is normally located in the **D:\TE1014\XBridge Base Install** directory on the CD. Select the sub-directory relevant to your operating system (Win9x, WinNT or Win2000). This will begin the installation of the software that locates and registers the bridge within the system registry. Follow the instructions on the screen to complete installation. If no CD drive is present on the system, copy the relevant files from the CD onto a floppy disk.

4.2 Option Driver Installation

Insert the CD supplied with the unit into a CD drive that is accessible by the PXI Controller used with the chassis. Locate and run the SETUP.EXE program. This software is normally located in the **D:\TE1014\Test Panel** directory. This will begin the installation of the software applications and drivers that are used to allow access to the system monitoring. Follow the instructions on the screen to complete installation.

Note: If no CD drive is present on the system, copy the relevant folder onto a floppy disk and run as explained above.

4.3 Installing other PXI/CompactPCI modules

Installation of additional PXI/CompactPCI modules should be carried out as per the individual instructions for each PXI/CompactPCI module.

4.4 Verifying Installations

4.4.1 Bridge Card

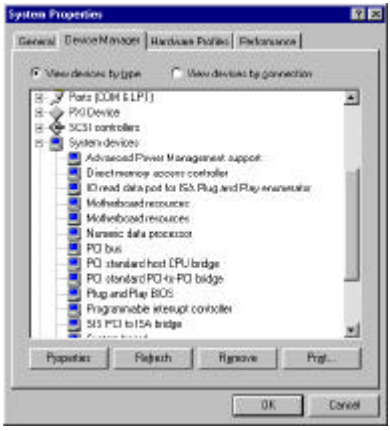


Figure 4.2 - Registered Devices.

To verify that the bridge card is operating correctly with the Operating System, open the control panel window, select system and then click on the Device manager tab. You should see the PCI to PCI Bridge listed as shown in figure 4.2 above. (You may have to expand the Systems Devices tree.) In some systems this will also be identified with the reference 21150.

4.4.2 Other PXI Modules

Some PXI Modules will be supplied with their own test/verification program. However, the National Instruments software application, VISA Interactive Control can also be used to verify PXI module installation. This is a program, which identifies installed VISA resources including PXI resources as shown in figure 4.3 below.

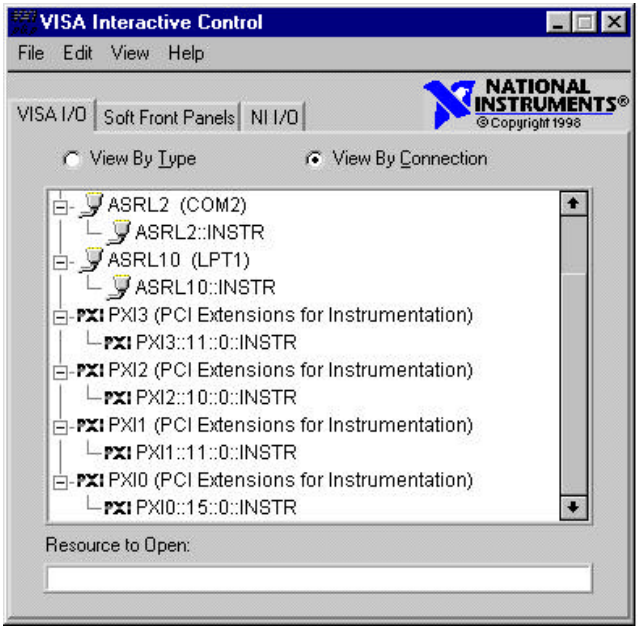


Figure 4.3 - VISA Resources.

In this example, there are three different PXI modules installed. These identification numbers identify which slot the module is installed in. Slot 1 is always the controller slot. Slot 2 is either the star trigger slot or a

general peripheral slot. Module slots 3 to 14 are all general peripheral slots. Each module slot has a unique identifier. These identifiers are shown in table 4.1 below.

Instrument identifiers which begin with 'PXI0' are normally contained within the first bus segment. Instrument identifiers which begin with 'PXI1' are normally contained within the second bus segment. If additional chassis are linked with a 'Standard' PCI to CompactPCI/PXI bridge, each bus segment should be numbered sequentially (e.g. PXI2, PXI3, etc.).

VISA Instrument Description	Slot Number
PXI0::20::0::INSTR[Note 1]	Slot 2
PXI0::19::0::INSTR	Slot 3
PXI0::18::0::INSTR	Slot 4
PXI0::16::0::INSTR	Slot 5
PXI0::15::0::INSTR	Slot 6
PXI0::14::0::INSTR	Slot 7
PXI1::15::0::INSTR	Slot 8
PXI1::14::0::INSTR	Slot 9
PXI1::13::0::INSTR	Slot 10
PXI1::12::0::INSTR	Slot 11
PXI1::11::0::INSTR	Slot 12
PXI1::10::0::INSTR	Slot 13
PXI1::09::0::INSTR	Slot 14

Table 4.1

Note1: For VISA Interactive Control Version 2.5 or greater, the VISA description has slightly changed. e.g. ***PXI0::20::INSTR***.

Bus-mastering is supported on the primary segment of the backplane and on the secondary segment of the backplane. Inter-segment bus-mastering is also supported between the primary and secondary bus. Trigger Bus X-Point connection is also supported.

5 Operation and Connectivity

5.1 TE1014

5.1.1 Chassis topology

The TE1014 PXI chassis is the standard model of chassis. It allows the insertion and use of one controller, Star Trigger, and up to thirteen CompactPCI/PXI cards. However, the chassis can be used in a number of other configurations.

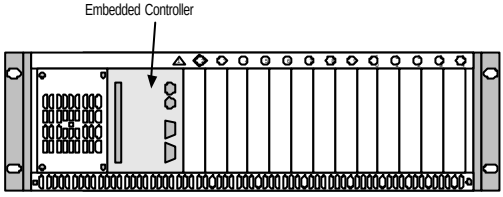


Figure 5.1 - Single Chassis with embedded Controller.

Figure 5.1 shows the chassis when used as a stand-alone unit with an embedded PXI controller in the controller slot. This configuration will

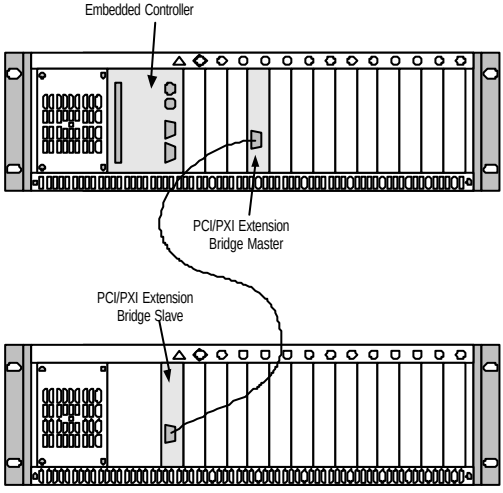


Figure 5.2 - Master and slave configuration.

take up to 13 peripheral cards. The primary bus will normally be designated as PXI0 and the secondary bus will be designated as PXI1. Figure 5.2 shows a Master/Slave configuration, where the second chassis is connected to the Primary chassis via a CompactPCI/PXI to CompactPCI/PXI extension bridge. In this instance, the primary bus of the master chassis will normally be designated as PXI0 and the secondary bus of the master chassis designated as PX1. The primary bus of the slave chassis will be designated as PXI2 and the secondary bus of the slave chassis will be designated as PXI3. The configuration shown in Figure 5.2 gives the user the option to have up to 25 CompactPCI/PXI peripheral cards in one system. It is possible to use this type of Bridge technology to add additional chassis to a single system.

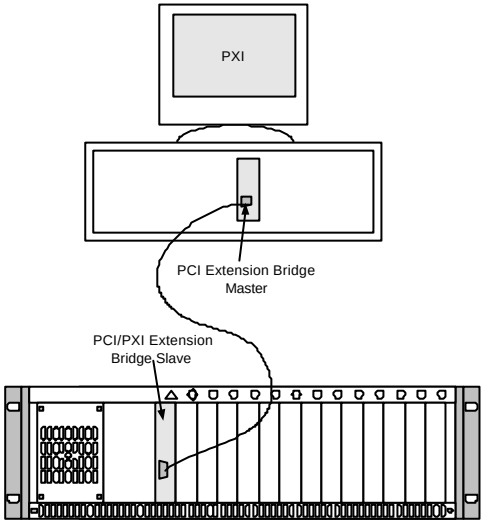


Figure 5.3 - Desktop PC (PCI expansion Bridge) to PXI Chassis.

The example in figure 5.3 shows how a standard Desktop PC can be used to control a PXI chassis by using a PCI to CompactPCI/PXI extension bridge. In some instances, depending on the hardware configuration of the PC and the configuration of its BIOS, the primary bus in the PXI chassis could be identified as PXI1 while the secondary bus would be identified as PXI2.

5.1.2 Peripheral slot identification

Each TE1014 chassis has 14 PXI slot locations. These are identified by numbers on the top section of the chassis. The first location is reserved for the controller. This can be either an embedded controller or a bridge expansion controller card. In a PXI chassis, this slot is always located in the left most slot of the PXI bus. It is always identified with the triangle glyph as shown in figure 5.4. and numbered '1'.

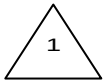


Figure 5.4 - Controller slot glyph.

The second slot position is the star trigger/peripheral card position. This can be used with any standard CompactPCI/PXI card or a special star trigger card. This slot is identified with a circle inside a diamond as shown in figure 5.5. It is numbered '2'.

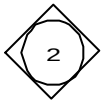


Figure 5.5 - Star trigger/general peripheral slot glyph.

Slots three to fourteen are all general peripheral slots. These slots are identified by the numbers '3' to '14' inclusive inside a circle.

Figure 5.6 - General peripheral slot glyph.

The fourteen slot positions are achieved by bridging one primary 8 slot PXI bus to a secondary 8 slot PXI bus. The bridge card occupies one slot position on each bus. The boundary between primary and secondary bus on the chassis is identified by the boundary glyph as shown in figure 5.7 below.

Figure 5.7 - Bus boundary glyph.

Each peripheral slot is identified in software with a unique number. When using the National Instruments software application, VISA Interactive Control, the slot identification of all PXI resources can be displayed as shown in figure 5.8 below.

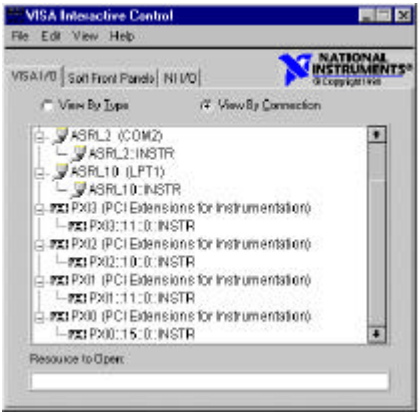


Figure 5.8 - PXI Resources.

Each PXI bus segment has a unique identifier per system. When using the TE1014 series of PXI chassis, the slots are identified as shown in table 5.1. The character 'a' represents the number that identifies the primary bus and the character 'b' represents the number that identifies the secondary bus.

VISA Instrument Description	Slot Number
PXIa::20::INSTR	Slot 2
PXIa::19::INSTR	Slot 3
PXIa::18::INSTR	Slot 4
17 Reserved for Bridge	N/A
PXIa::16::INSTR	Slot 5
PXIa::15::INSTR	Slot 6
PXIa::14::INSTR	Slot 7
PXIb::15::INSTR	Slot 8
PXIb::14::INSTR	Slot 9
PXIb::13::INSTR	Slot 10
PXIb::12::INSTR	Slot 11
PXIb::11::INSTR	Slot 12
PXIb::10::INSTR	Slot 13
PXIb::09::INSTR	Slot 14

Table 5.1

5.1.3 Star trigger

The PXI star trigger bus originates in the slot 2 position and is used to provide synchronization features to users of PXI systems. The star trigger bus provides a dedicated trigger line between slot 2 and each of the other 12 peripheral slots.

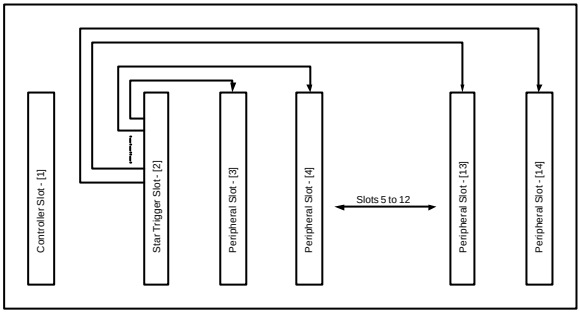

Controller Slot - [1]
Star Trigger Slot - [2]
Peripheral Slot - [3]
Peripheral Slot - [4]
Slots 5 to 12
Peripheral Slot - [13]
Peripheral Slot - [14]

Figure 5.9 - Star Trigger Routing.

This can provide very precise trigger signals to other peripheral modules. Each star trigger line is designed to provide an equal propagation delay and low skew from slot 2 to each of the other peripheral slots.

PXI Star Trigger slot Signal Name	Peripheral Slot Destination
PXI_STAR0	SLOT 3
PXI_STAR1	SLOT 4
PXI_STAR2	SLOT 5
PXI_STAR3	SLOT 6
PXI_STAR4	SLOT 7
PXI_STAR5	SLOT 8
PXI_STAR6	SLOT 9
PXI_STAR7	SLOT 10
PXI_STAR8	SLOT 11
PXI_STAR9	SLOT 12
PXI_STAR10	SLOT 13
PXI_STAR11	SLOT 14

Figure 5.10 - Star trigger routing from slot 2.

Star trigger signals are identified as PXI_STAR0 to PXI_STAR11. See appendix E (page 56) for slot pin details.

5.1.4 Local Bus

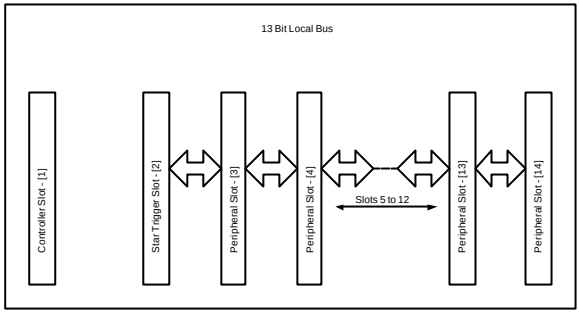

13 Bit Local Bus
Controller Slot - [1]
Star Trigge Slot - [2]
Peripheral Slot - [3]
Peripheral Slot - [4]
Slots 5 to 12
Peripheral Slot - [13]
Peripheral Slot - [14]

Figure 5.11 - Local Bus.

A 13 bit local bus is provided between each peripheral card position on the PXI backplane. This bus can be used to provide a data path between any peripheral card and the peripheral card immediately adjacent to it without affecting the bandwidth of the PCI Bus. The local bus to the left of a card is called the Local Bus Left (LBL) and the local bus to the right of a card is called the Local Bus Right (LBR). There is no local bus to the left of the star trigger slot (slot 2). The local bus can be used for signals with a level of up to 42 Volts. Local bus signals are identified as PXI_LBL0 to PXI_LBL12 and PXI_LBR0 to PXI_LBR12.

5.1.5 PXI Trigger Bus

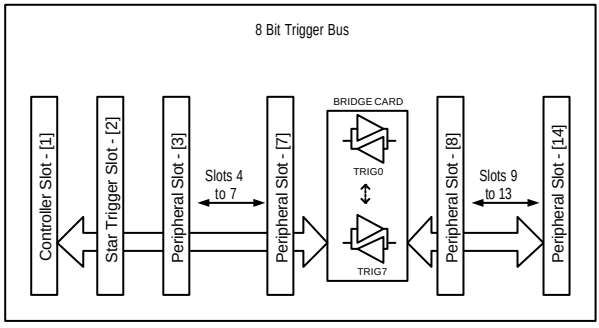


Figure 5.12 - 8 bit trigger bus.

The PXI trigger bus provides an 8-bit bus, which can be used for inter-module triggering, synchronization, data communication or clock transmission. The trigger bus bits are identified as PXI_TRIG0 to PXI_TRIG7.

On the TE1014 chassis, there are two PXI trigger busses implemented. One is on the primary side of the PXI bus and incorporates slots 1 (controller slot) to 7. The other is on the secondary PXI bus side and incorporates slots 8 to 14. Between the two PXI trigger busses is a matrix connected to bi-directional buffers and inverters. These buffers and inverters route the trigger bus signals from one side of the bus and can route it to any of the 8 trigger bus signals on the other side. It can also route the signal to another trigger bus signal on the same segment. Each signal can also be inverted.

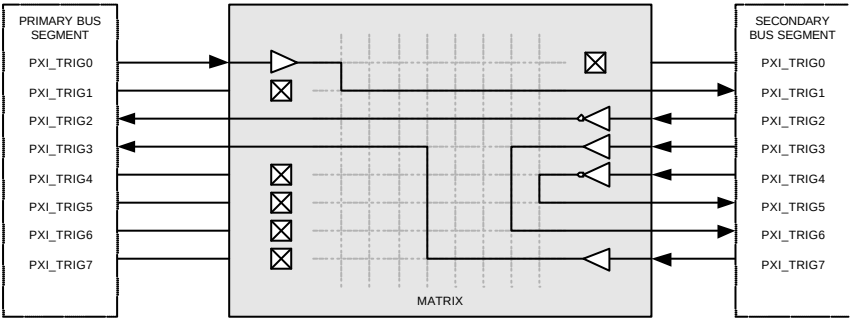


Figure 5.13 - Example of Bridge matrix routing.

5.1.6 10MHz system clock

The PXI system has a 10MHz system clock that is used to control inter-module timing. This clock has low skew properties and is ideal as a reference for synchronization between PXI modules. This clock is distributed to each peripheral slot and is called PXI_CLK10.

In the slot 2 (Star trigger/peripheral) position, there is a system clock input pin called PXI_CLK10_IN. On the TE1014 chassis, users can supply their own clock reference to this pin. A clock input on this pin will switch off the system 10MHz clock signal and replaces it.

5.1.7 Chassis Monitor Panel

The TE1014 chassis has voltage and temperature monitoring circuitry and PXI trigger bus buffer control. The users have available to them, software calls that give them access to these functions.

Supplied with the TE1014 is a software program that opens a sample control panel. This panel program allows the user to look at the voltage and temperature monitoring system and operate the PXI trigger buffers.

From the START menu, PROGRAMS, locate the PXI 14 Slot Team executable program and run it. This will display a panel as shown in figure 5.14 below. This panel allows the user to select between controls and indicators.

It also allows the user to select which bridge they want to talk to if more than one bridge is used in a system with multiple chassis. This is done by entering or selecting from the list box the correct Bus ID and Bridge ID relevant to the system. Selecting the <controls> button will display a trigger buffer control panel as shown in figure 5.15. The user can select each individual buffer and decide which direction the data should pass.

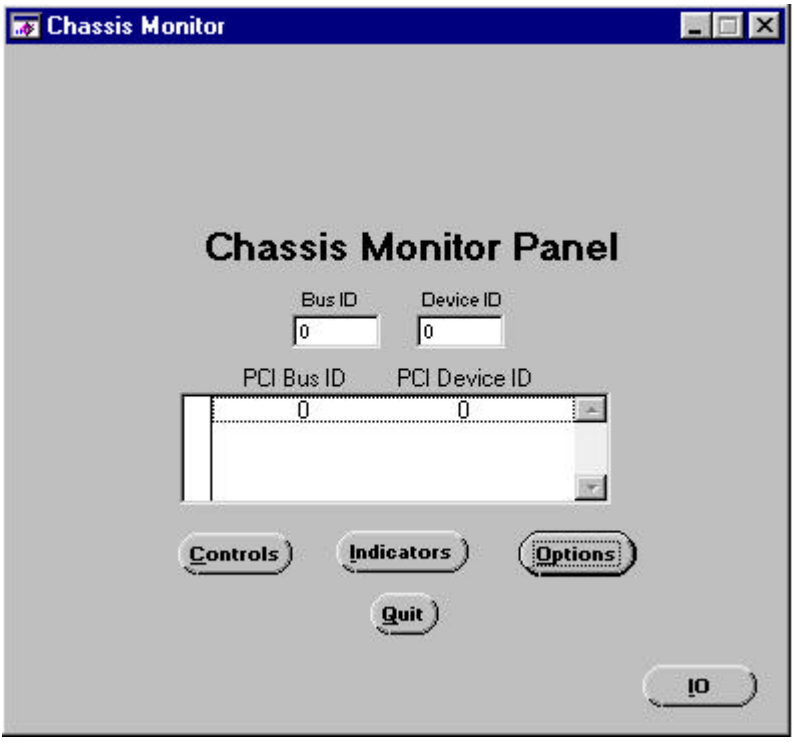


Figure 5.14 – TEAM Chassis Monitor Panel.

The user can also select how to route the trigger bus signal. If a trigger bus signal is routed back onto itself, it will provide a logic '1' or a logic '0' depending on whether a buffer or inverter is selected. If the trigger bus lines are 'not connected' the buffer/inverters are held in tri-state mode and no data can pass between either bus segment of the chassis on the tri-stated lines.

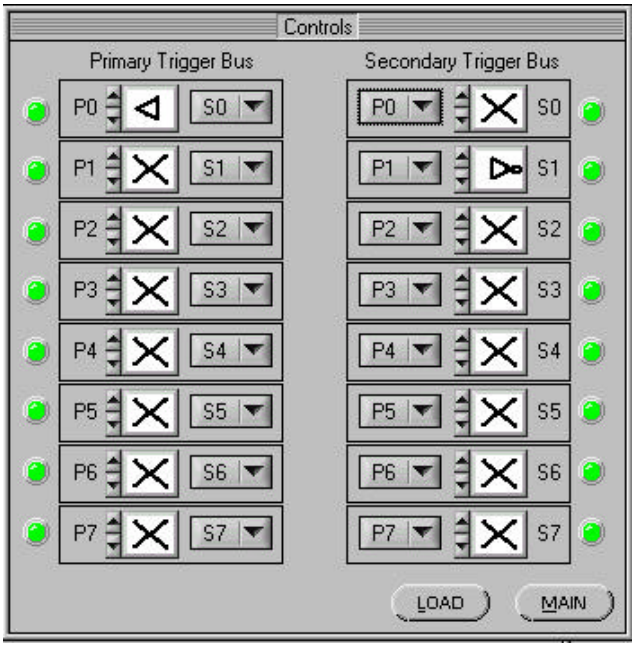


Figure 5.15 - Trigger Bus control panel.

Figure 5.15 shows an example of PXI_TRIG0 being routed from the secondary side of the bus to the primary side of the bus and PXI_TRIG1 being routed from the primary side of the trigger bus and being inverted onto the secondary trigger bus. After making changes, select the <LOAD> button to commit the changes.

The second panel available is the indicators panel as shown in figure 5.16 below. This panel is used to monitor the four main power supplies of the TE1014 chassis. These are the +5 Volts DC, the +3.3 Volts DC, the +12 Volts DC and the –12 Volts DC. The panel also displays indicators that monitor the chassis internal temperature.

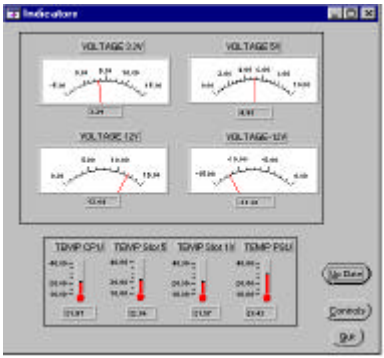


Figure 5.16 - Indicators Panel

There are sensors mounted over the CPU slot, Slot 5, Slot10 and one in the PSU compartment. The panel requires manual update by clicking on the ***<UpDate>*** button. The ***<Quit>*** will exit the Program.

There are two other temperature sensors in the chassis that control the speed of the fans. As the internal temperature of the chassis increases, the fan speed increases. These additional sensors are not monitored.

6 Appendices

Appendix A – TeamBridge.DLL Calls

TeamBridgeReadPCICount

Description

This function interrogates the Team module and identifies the module type.

C Prototype

```
ViInt16_VI_FUNC TeamBridgeReadPCICount();
```

Parameters

Input	Description	Data Type
Void	No value	Void

Return Values

Output	Description	Data Type
ViSuccess	Success.	ViInt16
-1	No bridge card info available, unable to use Teamcard.ini.	ViInt16
-2	Not a Teamcard type	ViInt16
-3	Relay parameters incorrect or not a valid relay card	ViInt16
-4	IO error	ViInt16

TeamBridgeReturn

Description

This function is a dummy call that returns the value that is passed to it.

C Prototype

```
ViInt16 _VI_FUNC TeamBridgeReturn(ViInt16 value);
```

Parameters

Input	Description	Data Type
Value	Any 16bit value.	ViInt16

Return Values

Output	Description	Data Type
Value	Should be the same as value passes	ViInt16
-1	Error	ViInt16

TeamBridgeInfo

Descr iption

This function returns a point to the Bridge map structure that is set up at the start of the initialisation stage. It contains all the information of any Team bridge in the system. Structure can be seen in the TeamBridge.h file.

C Prototype

```
Bridge_Map* _VI_FUNC TeamBridgeInfo(ViInt16 BusID, ViInt16 DevID);
```

Parameters

Input	Description	Data Type
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
Value	Pointer to bridge structure	BridgeMap* (ViInt16)
-1	Error	BridgeMap* (ViInt16)

TeamBridgeDisablePort

Description

This closes down communication to an individual bridge.

C Prototype

```
ViInt16 _VI_FUNC TeamBridgeDisablePort(ViInt16 BusID, ViInt16
DevID);
```

Parameters

Input	Description	Data Type
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
Value	PCI Device number.	ViInt16
-1	Error.	ViInt16

TeamBridgeEnablePort

Description

This function set up communication to the Team Bridge that you would like to commutate with.

C Prototype

```
ViInt16 _VI_FUNC TeamBridgeEnablePort(ViInt16 BusID, ViInt16
DevID);
```

Parameters

Input	Description	Data Type
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
ViSuccess	Port to bridge open success	ViInt16
-1	Error. No known bridge device in system	ViInt16

TeamVerfiyBridge

Description

This function looks in the bridge structure to check that there is a Team bridge in the system and that communication can take place with this device.

C Prototype

```
ViInt16 _VI_FUNC TeamBridgeVerfiyBridge(ViInt16 BusID, ViInt16
DevID);
```

Parameters

Input	Description	Data Type
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
ViSuccess	Team type bridge exists.	ViInt16
-1	Error	ViInt16

TeamBridgeReadVersion

Description

This function reads the firmware version of the embedded PIC code on the Team Bridge that you are communication with.

C Prototype

```
ViReal64 _VI_FUNC TeamBridgeReadVersion(ViInt16 BusID, ViInt16
DevID);
```

Parameters

Input	Description	Data Type
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
Value	PCI Device number	ViReal64
-1	Error	ViReal64 (ViInt16)

TeamBridgeWriteBuffer

Description

This function writes out an 8-bit word to one of two registers (0 and1). This registers control the bi-directional buffers that pass the PXI Triggering across the bridge.
Note! NOT used for X-Point Bridges

C Prototype

```
ViInt16 _VI_FUNC TeamBridgeWriteBuffer(ViInt16 regnumber, ViInt16
value, ViInt16 BusID, ViInt16 DevID);
```

Parameters

Input	Description	Data Type
Regnumber	Register address to write to.	ViInt16
Value	8bit Value to be written to a register.	ViInt16
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
ViSuccess	The current Team card version	ViInt16
-1	Error	ViInt16

TeamBridgeReadVoltage

Description

This function reads that voltage channels (0-3) in Volts. Channel 0= +12 volts, channel 1 = +5 volts, channel 2 = 3.3 volts, channel 3 = -12volts.

C Prototype

```
ViReal64 _VI_FUNC TeamBridgeReadVoltage(ViInt16 channel, ViInt16
BusID, ViInt16 DevID);
```

Parameters

Input	Description	Data Type
Channel	Voltage channel to read value	ViInt16
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
Value	Voltage read correctly in Volts.	ViReal64
-1	Facility not enabled	ViReal64 (ViInt16)

TeamBridgeReadTemperature

Description

This function reads temperature sensors (0-3) in degrees. Sensor 0- controller slot, sensor 1 = slot 5, sensor 2 = slot 10, sensor 3 = power supply bay.

C Prototype

```
ViReal64 _VI_FUNC TeamBridgeReadTemperature(ViInt16 sensor,
ViInt16 BusID, ViInt16 DevID);
```

Parameters

Input	Description	Data Type
Sensor	Temperature sensor to read value	ViInt16
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
Value	Temperature read correctly in degrees Celsius	ViReal64
-1	Facility not enables	ViReal64 (ViInt16)
-3	Error opening device	ViReal64 (ViInt16)

TeamBridgeWriteReg

Description

This function writes a value to one of the 64 registers on the bridge card.

C Prototype

```
ViInt16 _VI_FUNC TeamBridgeWriteReg(ViInt16 regnumber, ViInt16
value, ViInt16 BusID, ViInt16 DevID);
```

Parameters

Input	Description	Data Type
Regnumber	Register address to write to.	ViInt16
Value	8bit Value to be writhen to the register.	ViInt16
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
ViSuccess	Voltage read correctly	ViInt16
-1	Error	ViInt16

TeamBridgeReadReg

Description

This function reads a value from one of the 64 registers on the bridge card.

C Prototype

```
ViInt16 _VI_FUNC TeamBridgeReadReg(ViInt16 regnumber, ViInt16
BusID, ViInt16 DevID);
```

Parameters

Input	Description	Data Type
Regnumber	Register address to write to.	ViInt16
Value	8bit Value to writhen to the register.	ViInt16
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
Value	Returned value read from resister.	ViInt16 (ViUInt8)
-1	Error	ViInt16

TeamBridgeReadAuxADC

Description

This function reads a value for the user analogy to digital converter on the bridge card.

C Prototype

```
ViRead64 _VI_FUNC TeamBridgeReadAuxADC(ViInt16 BusID, ViInt16
DevID);
```

Parameters

Input	Description	Data Type
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
Value	Return value read (0-5) volts.	ViReal64
-1	Error	ViReal64(ViInt16)

TeamBridgeWriteEEReg

Description

This function writes a value to the on board EEDATA.

C Prototype

```
ViInt16 _VI_FUNC TeamBridgeWriteEEReg(ViInt16 regnumber, ViInt16
value, ViInt16 BusID, ViInt16 DevID);
```

Parameters

Input	Description	Data Type
RegNumber	Register address to write to.	ViInt16
Value	8bit Value to be writhen to the register.	ViInt16
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
ViSuccess	Voltage read correctly	ViInt16
-1	Error	ViInt16

TeamBridgeReadEEReg

Description

This function reads a value from the on board EEDATA.

C Prototype

```
ViInt16 _VI_FUNC TeamBridgeReadEEReg(ViInt16 regnumber, ViInt16
BusID, ViInt16 DevID);
```

Parameters

Input	Description	Data Type
ReNumber	Register address to write to.	ViInt16
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
Value	Data read from EEData	ViInt16
-1	Error	ViInt16

TeamWrite_To_Internal_DAC_Dual

Description

This function writes out the scaling values to the 2 Internal PSU's.
Note! Rev B Model

C Prototype

```
ViInt16 _VI_FUNC TeamWrite_To_Internal_DAC_Dual(ViInt16 BusID,
ViInt16 DevID, float Source1_scale, float Sourse2_scale);
```

Parameters

Input	Description	Data Type
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16
Source1_scale	Scaling value for the first power supply	Float
Source2_scale	Scaling value for the second power supply	Float

Return Values

Output	Description	Data Type
Value	PCI Device number	ViInt16
-1	Error	ViInt16

TeamWrite_To_Internal_DAC_Quad

Description

This function writes out the scaling values to the 4 Internal PSU's.
Note! Rev B Model

C Prototype

```
ViInt16 _VI_FUNC TeamWrite_To_Internal_DAC_Quad(ViInt16 BusID,
ViInt16 DevID, float Source1_scale, float Sourse2_scale,float
Source3_scale, float Sourse4_scale);
```

Parameters

Input	Description	Data Type
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16
Source1_scale	Scaling value for the first power supply	Float
Source2_scale	Scaling value for the second power supply	Float
Source3_scale	Scaling value for the third power supply	Float
Source4_scale	Scaling value for the forth power supply	Float

Return Values

Output	Description	Data Type
Value	PCI Device number	ViInt16
-1	Error	ViInt16

TeamWrite_To_External_DAC_Quad

Description

This function writes out the scaling values to the 4 External PSU's.
Note! Rev B Model

C Prototype

```
ViInt16 _VI_FUNC TeamWrite_To_External_DAC_Quad(ViInt16 BusID,
ViInt16 DevID, float Source1_scale, float Sourse2_scale,  float
Source3_scale, float Sourse4_scale);
```

Parameters

Input	Description	Data Type
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16
Source1_scale	Scaling value for the first power supply	Float
Source2_scale	Scaling value for the second power supply	Float

Return Values

Output	Description	Data Type
Value	PCI Device number	ViInt16
-1	Error	ViInt16

TeamReadOpto_IO_Port

Description

This function reads in data from the optically isolated port on the B- option chassis.
Note! Rev B Model

C Prototype

```
ViInt16 _VI_FUNC TeamReadOpto_IO_Port(ViInt16 BusID, ViInt16
DevID);
```

Parameters

Input	Description	Data Type
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16

Return Values

Output	Description	Data Type
Value	Value from IO Port	ViInt16

TeamWriteOpto_IO_Port

Description

This function writes out date to the optically isolated port on the B- option chassis.

C Prototype

```
ViInt16 _VI_FUNC TeamWriteOpto_IO_Port(ViInt16 BusID, ViInt16
DevID, ViInt16 Data);
```

Parameters

Input	Description	Data Type
BusID	PCI Bus segment number in Hex.	ViInt16
DevID	PCI Device number in Hex.	ViInt16
Data	Data to be writhen to the Opto IO Port.	ViInt16

Return Values

Output	Description	Data Type
Value	PCI Device number	ViInt16
-1	Error	ViInt16

TeamCalculate_Scale_Values

Description

This function calculates the scaling values needed to control the PSU.
Note! Rev B Model

C Prototype

```
ViInt16 _VI_FUNC TeamCalculate_Scale_Values(float MaxLim, float
MinLim, float source);
```

Parameters

Input	Description	Data Type
MaxLim	Maximum limit for voltage sources	Float
MinLim	Minimum limit of voltage scourse	Float
Source	Voltage source to be controled	Float

Return Values

Output	Description	Data Type
Value	PCI Device number	ViInt16
-1	Error	ViInt16

TeamBridgeRouteTriggerBusChannel

Description

This function provides simple routing from A to B with inversion capability if required. For example, to connect TRIG3 (channel 3) on the primary side of the bus to TRIG1 on the secondary side of the bus (channel 9), TRIG1 on the secondary side being the driver, TeamBridgeRouteTriggerBusChannel(3,9,0,bus,dev) would be used.

		Channel Number:
Primary Bus	PXI_TRIG0	0
	PXI_TRIG1	1
	PXI_TRIG2	2
	PXI_TRIG3	3
	PXI_TRIG4	4
	PXI_TRIG5	5
	PXI_TRIG6	6
	PXI_TRIG7	7
Secondary Bus	PXI_TRIG0	8
	PXI_TRIG1	9
	PXI_TRIG2	10
	PXI_TRIG3	11
	PXI_TRIG4	12
	PXI_TRIG5	13
	PXI_TRIG6	14
	PXI_TRIG7	15

Multiple channels can be connected to a single source if required. A further option is to connect a channel to itself , which results in the channel driving either a 1 or 0 depending on the invert option.

Note! X-Point Bridge Only

C Prototype

```
ViInt16_VI_FUNC TeamBridgeRouteTriggerBusChannel(ViInt16
destinationchannel, ViInt16 sourcechannel, ViInt16 invert, ViInt16
BusID, ViInt16 DevID);Parameters
```

Input	Description	Data Type
`Destination channel`	The output channel to be used	`ViInt16`
`Source channel`	The input channel to be used	`ViInt16`

`Invert`	Invert the output if 1	`ViInt16`
`Bus ID`	The Bridge PCI Bus Number	`ViInt16`
`Dev ID`	The Bridge PCI device ID	`ViInt16`

Return Values

Output	Description	Data Type
ViSuccess	Success.	ViInt16
-1	No known bridge device, comms error	ViInt16
-2	Channel out of range	ViInt16

TeamBridgeDisableTriggerBusChannel

Description

This function disable a specific channel from being driven by another channel effectively turning the output off. The channel can of course still drive other channels.
Note! X-Point Bridge Only

C Prototype

```
ViInt16_VI_FUNC TeamBridgeDisableTriggerBusChannel(ViInt16
channel, ViInt16 BusID, ViInt16 DevID);
```

Parameters

Input	Description	Data Type
`channel`	The channel number	`ViInt16`
`Bus ID`	The Bridge PCI Bus Number	`ViInt16`
`Dev ID`	The Bridge PCI device ID	`ViInt16`

Return Values

Output	Description	Data Type
ViSuccess	Success.	ViInt16
-1	No known bridge device, comms error	ViInt16
-2	Channel out of range	ViInt16

TeamBridgeDriveTriggerBusChannel

Description

This function sets a specific channel to either drive high or a low. It is similiar to the use of using TeamBridgeRouteTriggerBusChannel(), except with identical source and destinations.
Note! X-Point Bridge Only

C Prototype

```
ViInt16_VI_FUNC TeamBridgeDriveTriggerBusChannel(ViInt16 channel,
ViInt16 invert, ViInt16 enable, ViInt16 BusID, ViInt16 DevID);
```

Parameters

Input	Description	Data Type
channel	The output channel to be used	ViInt16
Enable	Enable or disable the output	ViInt16
Invert	Invert the output if 1	ViInt16
Bus ID	The Bridge PCI Bus Number	ViInt16
Dev ID	The Bridge PCI device ID	ViInt16

Return Values

Output	Description	Data Type
ViSuccess	Success.	ViInt16
-1	No known bridge device, comms error	ViInt16
-2	Channel out of range	ViInt16

TeamBridgeReadBridgeIDs

Description
This function returns a list of PCI Bus and PCI Device numbers of all known PCI-PCI bridges

C Prototype

```
ViInt16_VI_FUNC TeamBridgeReadBridgeIDs(ViInt16 *BusID, ViInt16
*DevID, ViUInt8 *count);
```

Parameters

Input	Description	Data Type
count	On entry, A count describing how many array locations are available, and on exit how many bridges were returned	ViUInt8
Bus ID	An array of Bridge PCI Bus Numbers	ViInt16*
Dev ID	An array of Bridge PCI device IDs	ViInt16*

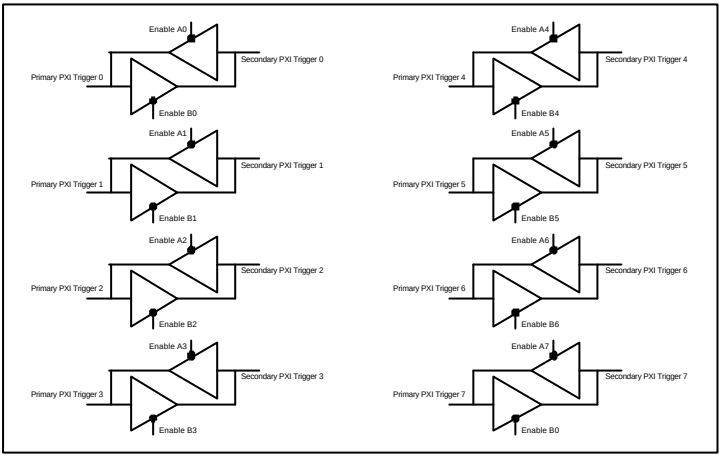
Note! count needs to be initialised to Array size.

Return Values

Output	Description	Data Type
ViSuccess	Success.	ViInt16
-1	Insufficient array size	ViInt16

Appendix B - PXI Trigger Buffer

Notes: This applies to NON X-Point Bridges.



To configure the PXI Trigger buffers from primary (controller side) to secondary bus segments. The user can configure the buffers directly from their own source code or with the user front panel supplied with the TE1014.

To configure using our direct .DLL call:

ViInt16 VI_FUNC TeamBridge WriteBuffer(ViInt16 regnumber, ViInt16 value);

Return Values:
-1 Error
ViSuccess Success

Description: This writes out an 8-bit word to one of two registers (0 and 1). The registers control the bi-directional buffers that relay the PXI Triggering across the bridge.

Register 0 controls Buffer 0, 1, 2 and 3.
Register 1 controls Buffer 4, 5, 6 and 7.

The format of bits

REGISTER 0								
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Buffer 0								
None							X	0
Pri – Sec							1	1
Sec – Pri							0	1
Buffer 1								
None					X	0		
Pri – Sec					1	1		
Sec – Pri					0	1		
Buffer 2								
None			X	0				
Pri – Sec			1	1				
Sec – Pri			0	1				
Buffer 3								
None	X	0						
Pri – Sec	1	1						
Sec – Pri	0	1						

REGISTOR 1								
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Buffer 4								
None							X	0
Pri – Sec							1	1
Sec – Pri							0	1
Buffer 5								
None					X	0		
Pri – Sec					1	1		
Sec – Pri					0	1		
Buffer 6								
None			X	0				
Pri – Sec			1	1				
Sec – Pri			0	1				
Buffer 7								
None	X	0						
Pri – Sec	1	1						
Sec – Pri	0	1						

6.3 Appendix C - Problem Solving.

If VISA does not see the installed module, the module must not be registering with the operating system.

On Windows 98/2000 check the Control Panel/System/Devices from the Start Menu window to verify that the modules are present as shown in figure 6.1.1 and 6.1.2 below.

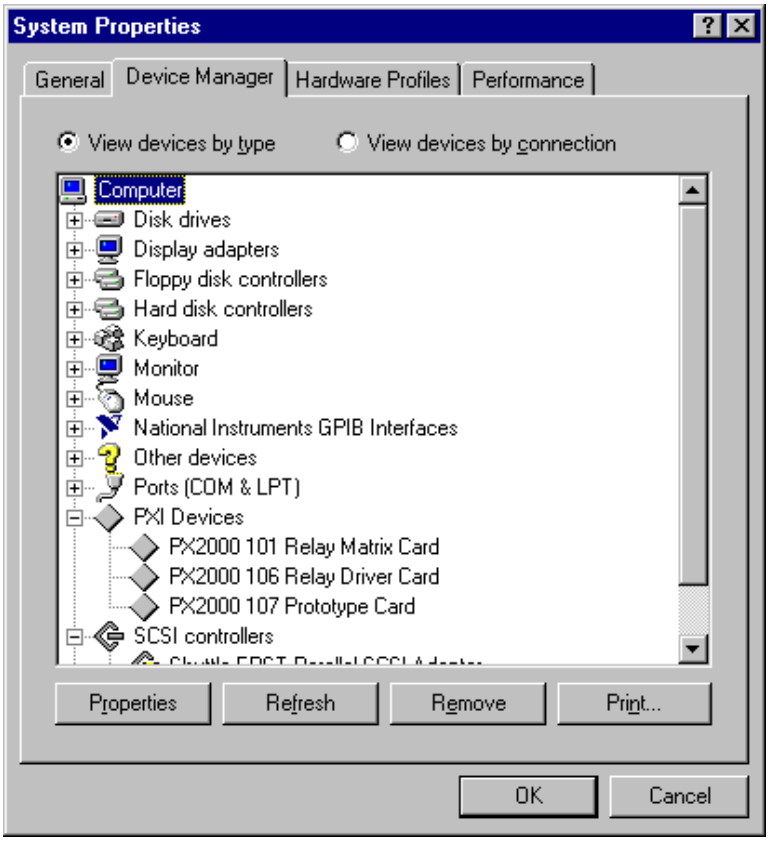


Figure 6.1.1 - Registered Devices

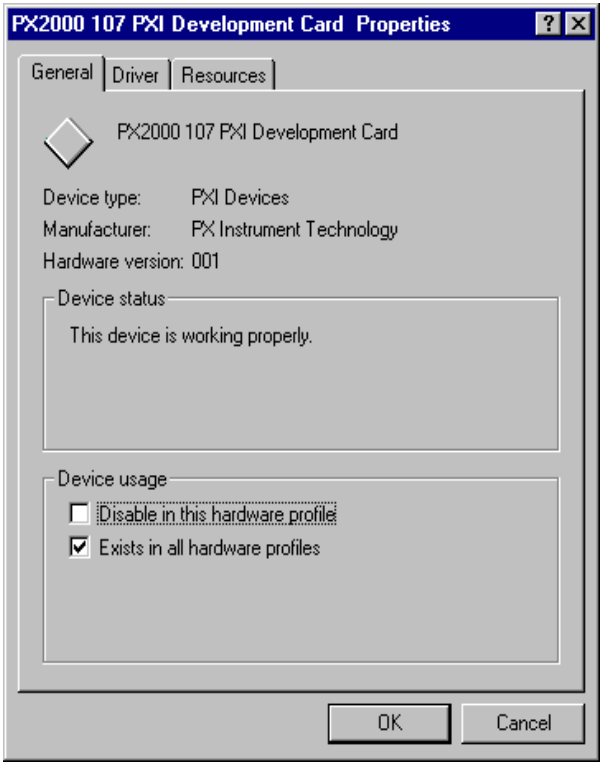


Figure 6.1.2 - Correctly configured device.

Check to see if an up to date version of NI_VISA is installed (must be 2.01 or greater).

Check that the relevant **.INF** file is in the Windows\INF directory as shown in Figure 6.1.3 below. (Win 98/2000, Win NT).

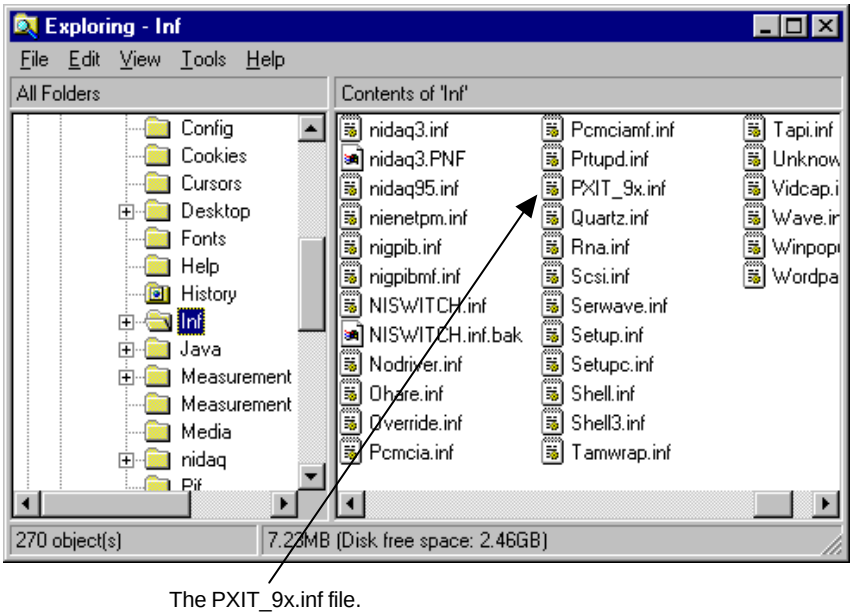


Figure 6.1.3 - INF Directory (Note: Example shows Windows

6.4 Appendix D - Example Program.

```
void main(void)
{
    /**Enable bridge to process commands**/
    if (TeamBridgeEnable() == 0)
    {
        printf("pci to pci bridge found\n");
        getchar();

        /**Read Chassis Voltage levels**/
        /**Print Values to screen*******/
        printf("+12V :%f \n",TeamBridgeReadVoltage(0));
        printf("+5V :%f \n",TeamBridgeReadVoltage(1));
        printf("+3.3V :%f \n",TeamBridgeReadVoltage(2));
        printf("-12V :%f \n",TeamBridgeReadVoltage(3));
        getchar();

        /**Read Chassis Temperatures**/
        /**Print Values to screen*****/
        printf("sensor #0 CPU :%f\n",TeamBridgeReadTemperature(0));
        printf("sensor #1 Slot 5 :%f\n",TeamBridgeReadTemperature(1));
        printf("sensor #2 Slot 10 :%f \n",TeamBridgeReadTemperature(2));
        printf("sensor #3 PSU :%f \n",TeamBridgeReadTemperature(3));
        getchar();

        /**Writes out a value to the registers**/
        /**Reads register value back***********/
        TeamBridgeWriteReg(0, 0xff);
        printf("0 0xff? %x\n",TeamBridgeReadReg(0));
        TeamBridgeWriteReg(0, 0x55);
        printf("0 0x55? %x\n",TeamBridgeReadReg(0));
        TeamBridgeWriteReg(1, 0xff);
        printf("1 0xff? %x\n",TeamBridgeReadReg(1));
        TeamBridgeWriteReg(1, 0x55);
        printf("1 0x55? %x\n",TeamBridgeReadReg(1));
        TeamBridgeWriteReg(2, 0xff);
        printf("2 0xff? %x\n",TeamBridgeReadReg(2));
        TeamBridgeWriteReg(2, 0x55);
        printf("2 0x55? %x\n",TeamBridgeReadReg(2));
        getchar();
    }

    /**Bridge failed to enable**/
    else
    {   printf("f ailed to start Bridge\n");
    }
    if (TeamBridgeEnable() == 0)
    {
        TeamBridgeDisable();
    }
    else
    {
        printf("failed to start Bridge\n");
    }

        /**End of program disable bridge**/
        TeamBridgeDisable();
```

6.5 Appendix E – Slot Pinout Table

Generic Peripheral Slot Pinout

22	GND	**GA4**	**GA3**	**GA2**	**GA1**	**GA0**	GND	
21	GND	**PXI_LBR0**	GND	**PXI_LBR1**	**PXI_LBR2**	**PXI_LBR3**	GND	
20	GND	**PXI_LBR4**	**PXI_LBR5**	**PXI_LBL0**	GND	**PXI_LBL1**	GND	
19	GND	**PXI_LBL2**	GND	**PXI_LBL3**	**PXI_LBL4**	**PXI_LBL5**	GND	
18	GND	**PXI_TRIG3**	**PXI_TRIG4**	**PXI_TRIG5**	GND	**PXI_TRIG6**	GND	P2
17	GND	**PXI_TRIG2**	GND	RSV	**PXI_STAR**	**PXI_CLK10**	GND	/
16	GND	**PXI_TRIG1**	**PXI_TRIG0**	RSV	GND	**PXI_TRIG7**	GND	J2
15	GND	**PXI_BRSVA15**	GND	RSV	**PXI_LBL6**	**PXI_LBR6**	GND	
14	GND	AD[35]	AD[34]	AD[33]	GND	AD[32]	GND	
13	GND	AD[38]	GND	V(I/O)	AD[37]	AD[36]	GND	C
12	GND	AD[42]	AD[41]	AD[40]	GND	AD[39]	GND	O
11	GND	AD[45]	GND	V(I/O)	AD[44]	AD[43]	GND	N
10	GND	AD[49]	AD[48]	AD[47]	GND	AD[46]	GND	N
9	GND	AD[52]	GND	V(I/O)	AD[51]	AD[50]	GND	E
8	GND	AD[56]	AD[55]	AD[54]	GND	AD[53]	GND	C
7	GND	AD[59]	GND	V(I/O)	AD[58]	AD[57]	GND	T
6	GND	AD[63]	AD[62]	AD[61]	GND	AD[60]	GND	O
5	GND	C/BE[5]#	GND	V(I/O)	C/BE[4]#	PAR64	GND	R
4	GND	V(I/O)	**PXI_BRSVB4**	C/BE[7]#	GND	C/BE[6]#	GND	
3	GND	**PXI_LBR7**	GND	**PXI_LBR8**	**PXI_LBR9**	**PXI_LBR10**	GND	
2	GND	**PXI_LBR11**	**PXI_LBR12**	UNC	**PXI_LBL7**	**PXI_LBL8**	GND	
1	GND	**PXI_LBL9**	GND	**PXI_LBL10**	**PXI_LBL11**	**PXI_LBL12**	GND	
25	GND	5V	REQ64#	ENUM#	3.3V	5V	GND	
24	GND	AD[1]	5V	V(I/O)	AD[0]	ACK64#	GND	
23	GND	3.3V	AD[4]	AD[3]	5V	AD[2]	GND	
22	GND	AD[7]	GND	3.3V	AD[6]	AD[5]	GND	
21	GND	3.3V	AD[9]	AD[8]	M66EN	C/BE[0]#	GND	
20	GND	AD[12]	GND	V(I/O)	AD[11]	AD[10]	GND	
19	GND	3.3V	AD[15]	AD[14]	GND	AD[13]	GND	P1
18	GND	SERR#	GND	3.3V	PAR	C/BE[1]#	GND	/
17	GND	3.3V	IPMB_SCL	IPMB_SDA	GND	PERR#	GND	J1
16	GND	DEVSEL#	GND	V(I/O)	STOP#	LOCK#	GND	
15	GND	3.3V	FRAME#	IRDY#	BD_SEL#	TRDY#	GND	
12–14	Key Area							C
11	GND	AD[18]	AD[17]	AD[16]	GND	C/BE[2]#	GND	O
10	GND	AD[21]	GND	3.3V	AD[20]	AD[19]	GND	N
9	GND	C/BE[3]#	IDSEL	AD[23]	GND	AD[22]	GND	N
8	GND	AD[26]	GND	V(I/O)	AD[25]	AD[24]	GND	E
7	GND	AD[30]	AD[29]	AD[28]	GND	AD[27]	GND	C
6	GND	REQ#	GND	3.3V	CLK	AD[31]	GND	T
5	GND	BRSVP1A5	BRSVP1B5	RST#	GND	GNT#	GND	O
4	GND	IPMB_PWR	GND	V(I/O)	INTP	INTS	GND	R
3	GND	INTA#	INTB#	INTC#	5V	INTD#	GND	
2	GND	TCK	5V	TMS	TDO	TDI	GND	
1	GND	5V	-12V	TRST#	+12V	5V	GND	
Pin	**Z**	**A**	**B**	**C**	**D**	**E**	**F**	

System Slot Pinout

22	GND	**GA4**	**GA3**	**GA2**	**GA1**	**GA0**	GND	
21	GND	CLK6	GND	RSV	RSV	RSV	GND	
20	GND	CLK5	GND	RSV	GND	RSV	GND	
19	GND	GND	GND	RSV	RSV	RSV	GND	
18	GND	**PXI_TRIG3**	**PXI_TRIG4**	**PXI_TRIG5**	GND	**PXI_TRIG6**	GND	P2
17	GND	**PXI_TRIG2**	GND	PRST#	REQ6#	GNT6#	GND	/
16	GND	**PXI_TRIG1**	**PXI_TRIG0**	DEG#	GND	**PXI_TRIG7**	GND	J2
15	GND	**PXI_BRSVA15**	GND	FAL#	REQ5#	GNT5#	GND	
14	GND	AD[35]	AD[34]	AD[33]	GND	AD[32]	GND	
13	GND	AD[38]	GND	V(I/O)	AD[37]	AD[36]	GND	C
12	GND	AD[42]	AD[41]	AD[40]	GND	AD[39]	GND	O
11	GND	AD[45]	GND	V(I/O)	AD[44]	AD[43]	GND	N
10	GND	AD[49]	AD[48]	AD[47]	GND	AD[46]	GND	N
9	GND	AD[52]	GND	V(I/O)	AD[51]	AD[50]	GND	E
8	GND	AD[56]	AD[55]	AD[54]	GND	AD[53]	GND	C
7	GND	AD[59]	GND	V(I/O)	AD[58]	AD[57]	GND	T
6	GND	AD[63]	AD[62]	AD[61]	GND	AD[60]	GND	O
5	GND	C/BE[5]#	GND	V(I/O)	C/BE[4]#	PAR64	GND	R
4	GND	V(I/O)	**PXI_BRSVB4**	C/BE[7]#	GND	C/BE[6]#	GND	
3	GND	CLK4	GND	GNT3#	REQ4#	GNT4#	GND	
2	GND	CLK2	CLK3	SYSEN#	GNT2#	REQ3#	GND	
1	GND	CLK1	GND	REQ1#	GNT1#	REQ2#	GND	
25	GND	5V	REQ64#	ENUM#	3.3V	5V	GND	
24	GND	AD[1]	5V	V(I/O)	AD[0]	ACK64#	GND	
23	GND	3.3V	AD[4]	AD[3]	5V	AD[2]	GND	
22	GND	AD[7]	GND	3.3V	AD[6]	AD[5]	GND	
21	GND	3.3V	AD[9]	AD[8]	M66EN	C/BE[0]#	GND	
20	GND	AD[12]	GND	V(I/O)	AD[11]	AD[10]	GND	
19	GND	3.3V	AD[15]	AD[14]	GND	AD[13]	GND	P1
18	GND	SERR#	GND	3.3V	PAR	C/BE[1]#	GND	/
17	GND	3.3V	IPMB_SCL	IPMB_SDA	GND	PERR#	GND	J1
16	GND	DEVSEL#	GND	V(I/O)	STOP#	LOCK#	GND	
15	GND	3.3V	FRAME#	IRDY#	GND	TRDY#	GND	
12–14	Key Area							C
11	GND	AD[18]	AD[17]	AD[16]	GND	C/BE[2]#	GND	O
10	GND	AD[21]	GND	3.3V	AD[20]	AD[19]	GND	N
9	GND	C/BE[3]#	IDSEL	AD[23]	GND	AD[22]	GND	N
8	GND	AD[26]	GND	V(I/O)	AD[25]	AD[24]	GND	E
7	GND	AD[30]	AD[29]	AD[28]	GND	AD[27]	GND	C
6	GND	REQ#	GND	3.3V	CLK	AD[31]	GND	T
5	GND	BRSVP1A5	BRSVP1B5	RST#	GND	GNT#	GND	O
4	GND	IPMB_PWR	GND	V(I/O)	INTP	INTS	GND	R
3	GND	INTA#	INTB#	INTC#	5V	INTD#	GND	
2	GND	TCK	5V	TMS	TDO	TDI	GND	
1	GND	5V	-12V	TRST#	+12V	5V	GND	
Pin	**Z**	**A**	**B**	**C**	**D**	**E**	**F**	

Star Trigger Slot Pinout

22	GND	**GA4**	**GA3**	**GA2**	**GA1**	**GA0**	GND	
21	GND	**PXI_LBR0**	GND	**PXI_LBR1**	**PXI_LBR2**	**PXI_LBR3**	GND	
20	GND	**PXI_LBR4**	**PXI_LBR5**	**PXI_STAR0**	GND	**PXI_STAR1**	GND	
19	GND	**PXI_STAR2**	GND	**PXI_STAR3**	**PXI_STAR4**	**PXI_STAR5**	GND	
18	GND	**PXI_TRIG3**	**PXI_TRIG4**	**PXI_TRIG5**	GND	**PXI_TRIG6**	GND	P2
17	GND	**PXI_TRIG2**	GND	RSV	**PXI_CLK10_IN**	**PXI_CLK10**	GND	/
16	GND	**PXI_TRIG1**	**PXI_TRIG0**	RSV	GND	**PXI_TRIG7**	GND	J2
15	GND	**PXI_BRSVA15**	GND	RSV	**PXI_STAR6**	**PXI_LBR6**	GND	
14	GND	AD[35]	AD[34]	AD[33]	GND	AD[32]	GND	
13	GND	AD[38]	GND	V(I/O)	AD[37]	AD[36]	GND	C
12	GND	AD[42]	AD[41]	AD[40]	GND	AD[39]	GND	O
11	GND	AD[45]	GND	V(I/O)	AD[44]	AD[43]	GND	N
10	GND	AD[49]	AD[48]	AD[47]	GND	AD[46]	GND	N
9	GND	AD[52]	GND	V(I/O)	AD[51]	AD[50]	GND	E
8	GND	AD[56]	AD[55]	AD[54]	GND	AD[53]	GND	C
7	GND	AD[59]	GND	V(I/O)	AD[58]	AD[57]	GND	T
6	GND	AD[63]	AD[62]	AD[61]	GND	AD[60]	GND	O
5	GND	C/BE[5]#	GND	V(I/O)	C/BE[4]#	PAR64	GND	R
4	GND	V(I/O)	**PXI_BRSVB4**	C/BE[7]#	GND	C/BE[6]#	GND	
3	GND	**PXI_LBR7**	GND	**PXI_LBR8**	**PXI_LBR9**	**PXI_LBR10**	GND	
2	GND	**PXI_LBR11**	**PXI_LBR12**	UNC	**PXI_STAR7**	**PXI_STAR8**	GND	
1	GND	**PXI_STAR9**	GND	**PXI_STAR10**	**PXI_STAR11**	**PXI_STAR12**	GND	
25	GND	5V	REQ64#	ENUM#	3.3V	5V	GND	
24	GND	AD[1]	5V	V(I/O)	AD[0]	ACK64#	GND	
23	GND	3.3V	AD[4]	AD[3]	5V	AD[2]	GND	
22	GND	AD[7]	GND	3.3V	AD[6]	AD[5]	GND	
21	GND	3.3V	AD[9]	AD[8]	M66EN	C/BE[0]#	GND	
20	GND	AD[12]	GND	V(I/O)	AD[11]	AD[10]	GND	
19	GND	3.3V	AD[15]	AD[14]	GND	AD[13]	GND	P1
18	GND	SERR#	GND	3.3V	PAR	C/BE[1]#	GND	/
17	GND	3.3V	IPMB_SCL	IPMB_SDA	GND	PERR#	GND	J1
16	GND	DEVSEL#	GND	V(I/O)	STOP#	LOCK#	GND	
15	GND	3.3V	FRAME#	IRDY#	BD_SEL#	TRDY#	GND	
12–14	Key Area							C
11	GND	AD[18]	AD[17]	AD[16]	GND	C/BE[2]#	GND	O
10	GND	AD[21]	GND	3.3V	AD[20]	AD[19]	GND	N
9	GND	C/BE[3]#	IDSEL	AD[23]	GND	AD[22]	GND	N
8	GND	AD[26]	GND	V(I/O)	AD[25]	AD[24]	GND	E
7	GND	AD[30]	AD[29]	AD[28]	GND	AD[27]	GND	C
6	GND	REQ#	GND	3.3V	CLK	AD[31]	GND	T
5	GND	BRSVP1A5	BRSVP1B5	RST#	GND	GNT#	GND	O
4	GND	IPMB_PWR	GND	V(I/O)	INTP	INTS	GND	R
3	GND	INTA#	INTB#	INTC#	5V	INTD#	GND	
2	GND	TCK	5V	TMS	TDO	TDI	GND	
1	GND	5V	-12V	TRST#	+12V	5V	GND	
Pin	**Z**	**A**	**B**	**C**	**D**	**E**	**F**	

7 Notes